

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of preventing buffer overrun security vulnerabilities comprising:
placing a return address on a stack;
adding a plurality of empty spaces to a known place on the stack;
executing a called function;
removing one or more of the plurality of empty spaces from the known place on the stack where they were previously placed to find the return address; and
setting an end of stack pointer to an end of stack frame.
2. (Previously Presented) The method of claim 1, further comprising:
calculating a random number;
saving the random number in a secure location;
placing a plurality of blank bytes equal to the random number to the stack;
building a stack frame by placing values from the called function to the stack; and
setting an end of stack pointer to an end of the stack frame.
3. (Previously Presented) The method of claim 2, wherein the location comprises a processor register that is not generally accessible.

4. (Previously Presented) The method of claim 1, further comprising:
recalling the random number saved;
removing a number of bytes equal to the random number from the stack;
retrieving the return address for the called function from the stack; and
setting an end of stack pointer to an end of a previous stack frame.
5. (Previously Presented) The method of claim 1, further comprising:
calculating a hash value of stack invariants;
saving the hash value in a secure location; and
building a stack frame by placing values from the called function onto the stack.

Claims 6-11 (Cancelled)

12. (Currently Amended) An apparatus comprising:
a storage device having stored therein one or more routines for preventing buffer
 overflow security vulnerabilities; and
a processor coupled to the storage device for executing the one or more routines
 that, when executing the routines, prevents buffer overflow errors by:
placing a return address on a stack;
adding a plurality of empty spaces to a known place on the stack;
executing a called function;
removing one or more of the plurality of empty spaces from the known place on
 the stack where they were previously placed to find the return address; and
setting an end of stack pointer to an end of stack frame.

13. (Previously Presented) The apparatus of claim 12, further comprising:
- calculating a random number;
 - saving the random number in a secure location;
 - placing a plurality of blank bytes equal to the random number to the stack;
 - building a stack frame by placing values from the called function onto the stack;
 - and
 - setting an end of stack pointer to an end of the stack frame.
14. (Previously Presented) The apparatus of claim 13, wherein location comprises a processor register that is not generally accessible.

Claims 15-22 (Cancelled)

23. (Currently Amended) A machine-readable medium having stored thereon data representing sets of instructions which, when executed by a machine, cause the machine to:
- place a return address on a stack;
 - adding a plurality of empty spaces to a known place on a stack;
 - execute a called function;
 - remove one or more of the plurality of empty spaces from the known place on the stack where they were previously placed to find the return address; and
 - set an end of stack pointer to an end of stack frame.

24. (Previously Presented) The machine-readable medium of claim 23, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
- calculating a random number;
 - saving the random number in a secure location;
 - placing a plurality of blank bytes equal to the random number to the stack;
 - building a stack frame by placing values from the called function onto the stack;
 - and
 - setting an end of stack pointer to an end of the stack frame.
25. (Previously Presented) The machine-readable medium of claim 24, wherein the location comprises a processor register that is not generally accessible.

Claims 26-33 (Cancelled)

34. (Currently Amended) A system, comprising:
- a storage medium; and
 - a processor coupled with the storage medium, the processor to
 - placing a return address on a stack,
 - adding a plurality of empty spaces to a known place on the stack,
 - executing a called function,
 - removing one or more of the plurality of empty spaces from the known place on the stack where they were previously placed to find the return address, and
 - setting an end of stack pointer to an end of stack frame.

35. (Previously Presented) The system of claim 34, further comprising:
- calculating a random number;
 - saving the random number in a secure location;
 - placing a plurality of blank bytes equal to the random number to the stack;
 - building a stack frame by placing values from the called function onto the stack;
 - and
 - setting an end of stack pointer to an end of the stack frame.
36. (Previously Presented) The system of claim 35, wherein location comprises a processor register that is not generally accessible.